面向软件定义互连系统的多协议交换电路

董春雷[1]，沈剑良[1]，李沛杰[1]，王盼[2]，薄光明[3]，路凯[1]
（1.信息工程大学，河南 郑州 450002；2.天津市滨海新区信息技术创新中心，天津 300457；
3.天津职业技术师范大学国有资产管理处，天津 300222）

摘　要：为满足软件定义互连系统中异构协议融合互连的需求，提出一种两级多协议交换电路，该电路通过融合共享缓存与Crossbar这2种交换架构，实现对多种异构协议的报文转发需求的兼顾。同时，提出一种基于时隙的多级仲裁调度方法，实现调度过程中时间与空间的解耦。仿真结果表明，所设计的交换电路能够弹性适应绑定模式变化引起的交换规模及端口缓存、端口带宽需求变化，缓存资源利用率相较传统组合输入输出排队（CIOQ）架构最高提高87.5%，转发时延低至数十纳秒，不仅适用于RapidIO、光纤通道（FC）、Ethernet、外设部件互连快速总线（PCIe）单一协议交换，而且适用于多种协议组合的混合协议交换。

关键词：软件定义互连；异构协议；交换结构；时分复用；通道绑定；缓存利用率
中图分类号：TP302
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024076

Multi-protocol switching circuit for software defined interconnection system

DONG Chunlei[1], SHEN Jianliang[1], LI Peijie[1], WANG Pan[2], BO Guangming[3], LU Kai[1]
1. Information Engineering University, Zhengzhou 450002, China
2. Information Technology Innovation Center of Tianjin Binhai New Area, Tianjin 300457, China
3. Department of State-owned Assets, Tianjin University of Technology and Education, Tianjin 300222, China

Abstract: In order to meet the requirements of heterogeneous protocol integration and interconnection in software-defined interconnection systems, a two-stage multi-protocol switching circuit was proposed. This circuit accommodated the forwarding needs of various heterogeneous protocols by integrating shared memory and Crossbar architectures. In addition, a multi-stage arbitrary scheduling scheme based on time slot was introduced to decouple time and space during the scheduling. Simulation results demonstrate that this switching circuit can dynamically adapt to changes in switching scale, port memory, and port bandwidth requirements resulting from binding mode changes. Compared to traditional combined input and output queued (CIOQ) architecture, the utilization of memory is increased by up to 87.5%, and the forwarding delay is as low as tens of nanoseconds, making it suitable for RapidIO, fibre channel (FC), Ethernet, peripheral component interconnect express (PCIe) single protocol switching as well as hybrid protocol switching combining these protocols.

Keywords: software defined interconnection, heterogeneous protocol, switching architecture, time division multiplexing, channel binding, memory utilization

收稿日期：2023–11–10；**修回日期：**2024–02–01
通信作者：沈剑良，jlshen1982@126.com
基金项目：国家重点研发计划基金资助项目（No.2022YFB2901000）
Foundation Item：The National Key Research and Development Program of China (No.2022YFB2901000)

0　引言

现有信息系统为支持不同应用场合的数据通信需求，通常采用多种异构互连协议，如中央处理器之间通常采用外设部件互连（PCI, peripheral component interconnect）总线和PCIe（PCI express）总线协议，信号处理器之间通常采用RapidIO协议，存储器之间通常采用光纤通道（FC, fibre channel）协议，内部网络通常采用Ethernet协议等。复杂信息系统或数据中心通常需要多个异构处理系统、存储系统和通信系统协同工作，不可避免地产生不同异构协议之间数据灵活交互的需求，当前通常采用单协议交换和多种桥接技术来满足异构协议的交互需求，此技术存在两方面的突出问题，一是需要采用多款芯片才能实现异构协议之间的互连互通，集成度低、复杂度高、功耗大；二是系统一旦部署完毕，系统结构及带宽分配随之固定，系统灵活性受限。基于软件定义互连的系统可以很好地解决上述效能与灵活性问题[1]。软件定义互连结构如图1所示，通常包括软件定义协议控制器、软件定义转发引擎和多协议交换电路3个部分。软件定义协议控制器可软件配置为支持不同协议、不同通道速率、不同绑定模式的协议控制器；软件定义转发引擎可软件配置为不同协议的解析与封装、不同协议的转换、不同协议的转发查表；多协议交换电路可实现不同协议类型、不同端口速率、不同交换规模数据的交换转发。多协议交换电路是软件定义互连的关键电路，其设计必须满足系统应用灵活性要求。不同协议对应的绑定模式及通道速率组合各异，对应端口数量、端口带宽和端口缓存需求各不相同，传统交换电路按照最大端口数与最大端口带宽、缓存进行设计即可满足要求，但这会造成极大的带宽与缓存资源浪费，如何做到能够弹性适应灵活多变的端口数量、端口带宽和端口缓存需求，进而保持较高的资源利用率，是多协议交换电路需要解决的难题。此外，不同协议的报文长度范围及对转发时延的要求也各不相同，能够消除大跨度报文长度对缓存利用率的影响，以及提供较低的转发时延也是多协议交换必须兼顾解决的问题。

当前学术界与产业界诸多设计对交换结构进行了研究[2-15]。文献[2]为提高常用的联合输入输出排队（CIOQ, combined input and output queuing）、联合输入交叉点排队（CICQ, combined input crosspoint queuing）交换架构的报文转发效率，提出了2种针对性的调度算法，但未解决Crossbar交换架构的端口特性不能随绑定模式弹性变化的灵活性问题，交换规模与端口速率随绑定模式变化时存在资源闲置现象。文献[3]针对卫星通信系统中星载交换电路硬件资源受限的问题，基于CICQ交换架构，通过多个端口共享内部总线，减少交叉节点个数的方式实现资源消耗的降低，但受限于交叉节点带缓存的交换结构，交叉节点数仍会随交换规模的增大而增多，对应缓存需求也会显著增加，因此，不适用于交换规模较大的场景。文献[4]针对时间敏感网络（TSN, time-sensitive networking）芯片最小化最大交换时延的需求，提出一种基于时分复用的多流水线交换架构及对应算法，但受限于共享缓存基本架构，不适用于RapidIO、PCIe等对转发时延要求较高的协议。IDT公司（已于2019年被瑞萨电子收购）针对基于Crossbar交换架构的RapidIO协议交换绑定模式变化时的资源闲置现象，在其推出的一款RapidIO高速交换芯片中提出了对应解决方案，通过端口缓存随绑定模式进行拆分与融合提高了不同绑定模式下的资源利用率，但应对的绑定模式有限，不适用于PCIe等具有8通道、16通道（8×、

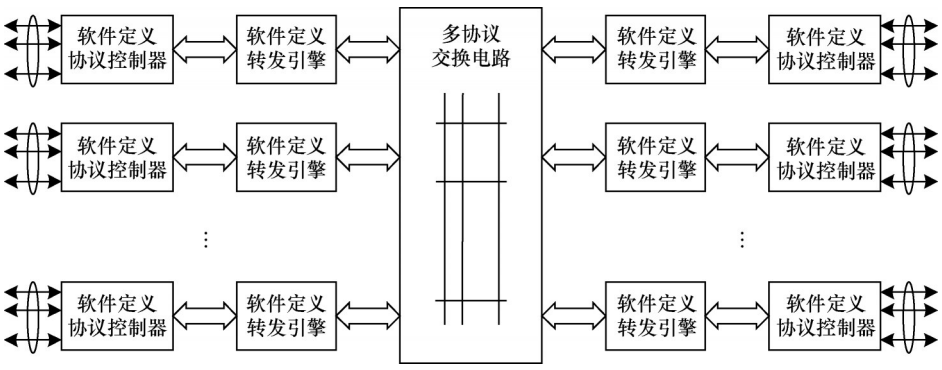


图1　软件定义互连结构

16×）绑定模式的协议。综上所述，已有研究大多是在特定应用协议前提下，针对交换电路的转发效率、资源消耗、转发时延等进行的突破或优化，适用于特定单一协议或某些具有相似交换需求的协议，而对于其他具有较大交换需求差异的协议则不适用或性能表现无法达到最优。为实现软件定义互连系统中多种异构协议的融合互连，迫切需要从支持不同交换特性的角度出发，研究能够兼顾多种异构协议的报文转发需求的多协议交换电路。

本文主要的研究工作如下。

1) 提出一种适用于软件定义互连系统的多协议交换架构。该架构以优势互补的方式融合共享缓存与Crossbar这2种主流交换架构，既具有共享缓存交换易于实现端口的绑定和拆分、能够适应端口数量和端口带宽变化的特性，又具有Crossbar交换的稳定性、良好的时延和抖动特性，从架构层面为统筹兼顾RapidIO 4.1、PCIe 4.0、16G FC及200G Ethernet这4种异构协议不同报文转发需求提供了可能。

2) 提出一种与多协议交换架构相适配的调度方法。多协议交换架构是一种新架构，需要为其设计一种兼顾设计复杂性与可实现性，同时能够提供最大转发效率的适配性调度方法。为此，基于时分与空分交织的架构特点，通过引入时隙的概念，实现了调度过程时间与空间的解耦，解决了交换电路的调度问题。

3) 对所提交换电路进行了前端实现与仿真验证，结果表明，本文设计的交换电路具有很好的灵活性与缓存资源利用率，并具有良好的时延性能，适用于多种异构协议。与传统交换电路及近年来有关参考设计对比，验证了本文设计的有效性。

1　多协议交换电路设计

1.1　设计需求分析

本文所设计的交换电路需要同时满足16G FC、RapidIO 4.1、PCIe 4.0、200G Ethernet协议的通道绑定模式、端口速率、报文长度以及交换时延等要求，如表1所示。

由表1可知，本文要支持的4种异构协议中除FC协议外，都支持不同的通道绑定模式，对应通道的绑定和拆分，从而导致交换端口数量和交换端口带宽需求的变化。高带宽对应高缓存，所以在交换端口带宽需求变化的同时，缓存需求也会变化。结合绑定模式下存在闲置端口的客观事实，为提高带宽、缓存资源利用率，端口间的缓存、带宽需要能够根据绑定模式动态分配或共享。多种异构协议的报文长度范围较大（8~4 096 B），为保障缓存利用率，多协议交换电路必须采用适当的缓存管理机制。为满足多种异构协议的时延指标要求，多协议交换电路必须具备低时延转发能力。多协议交换电路最终实现对4种单一协议交换及多种协议组合的混合协议交换的支持。

1.2　架构设计

通过研究共享总线交换、环形总线交换、共享缓存交换、Crossbar矩阵交换等主流交换架构，并对比分析RapidIO、FC、PCIe和Ethernet等典型协议交换芯片的实现架构，发现应用最广泛的2种交换架构为共享缓存交换和Crossbar矩阵交换，它们大量应用在单一协议交换中，但因各自的优缺点，无法很好地同时满足4种异构协议的报文转发需求。

共享缓存交换全端口时分复用的特性和后端实现时存储摆放的问题，不利于RapidIO和PCIe协议交换纳秒级转发时延的实现。Crossbar交换各个输出端口的转发总线相互独立，互不影响，这带来了良好的时延和抖动特性，但是这种交换架构的端口数量不具有弹性，且其对称性要求各个端口的带宽、缓存设置一致，对应交换芯片的最大端口数量和最大端口带宽，这会导致绑定模式下端口带宽、

表1　异构协议交换电路设计要求

协议	单通道速率/(Gbit·s^{-1})	通道绑定模式	端口速率/(Gbit·s^{-1})	报文长度/B	交换时延（64 B无阻塞）	交换架构应用场景
16G FC	4.25, 8.5, 14.025	1×	4.25~14.025	36~2 188	<1 μs	FC单一协议交换
RapidIO 4.1	1.25, 2.5, 3.125, 5, 6.25, 10.312 5, 12.5, 25.781 25	1×, 2×, 4×	1.25~103.125	8~280	<200 ns	RapidIO单一协议交换
PCIe 4.0	5, 8, 16	1×, 2×, 4×, 8×, 16×	5~256	12~2 068	<200 ns	PCIe单一协议交换
200G Ethernet	10.312 5, 25.781 25, 26.562 5, 53.125	1×, 2×, 4×	10.312 5~212.5	64~4 096	<2 μs	Ethernet单一协议交换

缓存资源的严重浪费。

如前所述，当前主流的交换架构无法直接作为多协议交换电路，因为多协议交换电路需要兼顾考虑各异构协议报文的交换特性和技术指标要求。

图2为多协议交换电路结构。全交换N个端口平均分为$\frac{N}{n}$个端口组（PG, port group），每个PG内的n个端口共享输入缓存和输出缓存，共享缓存结构构成第一级交换；$\frac{N}{n}$个PG通过$\frac{N}{n}\times\frac{N}{n}$的Crossbar交换连接，Crossbar为第二级交换。为实现低时延，Crossbar交换结构未设置交叉节点缓存，而是通过调度算法实现入口和出口的选通匹配。

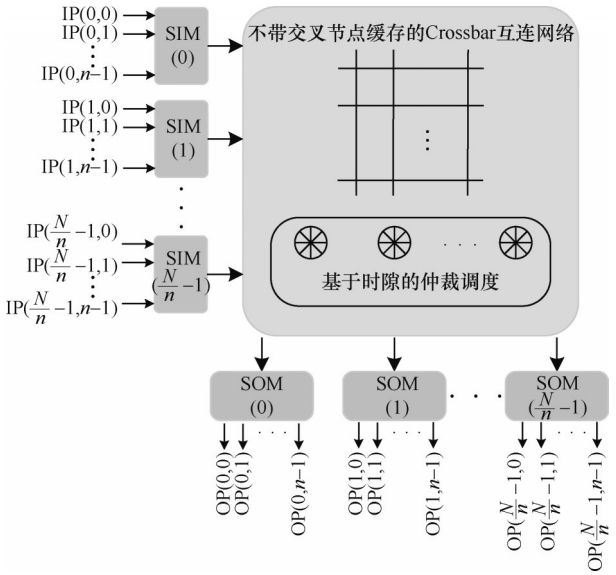


图2　多协议交换电路结构

整个交换电路包括共享缓存输入模块（SIM, shared-memory input module）、不带交叉节点缓存的Crossbar互连网络、基于时隙的仲裁调度和共享缓存输出模块（SOM, shared-memory output module）。

交换电路整体可看作CIOQ交换结构，PG内的共享输入和输出缓存可视作CIOQ交换结构的两级缓存，PG内的多个端口构成CIOQ交换结构的“一个”端口。为描述方便，将本文所提结构称为组合共享缓存输出队列（CSOQ, combined shared memory and output queuing）交换结构。

PG内的高速通道支持绑定，与之对应的交换端口也支持绑定，带来第一级交换结构内部缓存和带宽分配的弹性变化需求，这些弹性变化需求对于共享缓存交换结构来说是较易实现的。端口非绑定时（端口数量多、端口带宽小），共享缓存只拆分一部分给各端口使用；端口绑定时（端口数量少、端口带宽大），共享缓存则合并后使用，能够实现在不影响性能的前提下对缓存的最大程度利用。同时对于第二级交换结构来说，端口数量和带宽均未发生变化，保证了Crossbar交换结构的稳定性。

2　多协议交换关键电路及关键机制设计

2.1　共享缓存输入结构设计

SIM作为流量多路复用器，以时分复用的方式将n个端口接入Crossbar互连网络，其中n是由系统配置确定的常量。SIM结构如图3所示，其中，n=8，实线与虚线分别代表入队及出队相关数据与信号流，mem if和mem wif分别代表缓存接口和缓存写接口。从链路接收到的报文首先会存入为输入端口分配的输入缓存，之后经基于时隙的仲裁调度后通过Crossbar互连网络转发至对应的SOM。

为实现多个端口数据的并行处理，SIM为每个端口分配一套独立的处理逻辑，处理逻辑主要包括入队和出队2个部分，入队是指将前级模块输入的携带路由信息的报文及相关信息按照设定的存储粒度和管理方式存入缓存，并生成队列管理所需要的信息；出队是指调度器根据队列信息生成调度结果，指示传输控制模块将报文从对应的缓存中读出并发往后级，同时将占用的缓存资源进行释放。

绑定模式下，绑定端口的缓存需求增大，同时部分端口不再有效，其端口缓存暂时闲置，为充分利用缓存资源，缓存分配过程中会根据绑定模式将无效端口的缓存进行映射处理，从逻辑上实现与绑定端口的缓存的融合。

出队逻辑是SIM的关键处理逻辑，用于完成虚拟输出队列（VOQ, virtual output queue）的管理、调度，缓存的读取和完整总线数据的重构。SIM可以并行向后级传输n个不同报文，此种情况下需要实现对不同发送整包的跟踪，为此，SIM中设置n个出队控制子模块。

异构协议报文长度差异较大，为提高报文存储缓存的利用率，设计中采用基于cell（固定大小的数据单元）而非基于整包的缓存管理方式。缓存管理的最小单位为Datanode，报文以Datanode链表的形式存储在RAM中，Datanode的链表信息存储在Overhead信息中。Datanode和Overhead信息的数据结构如图4所示。

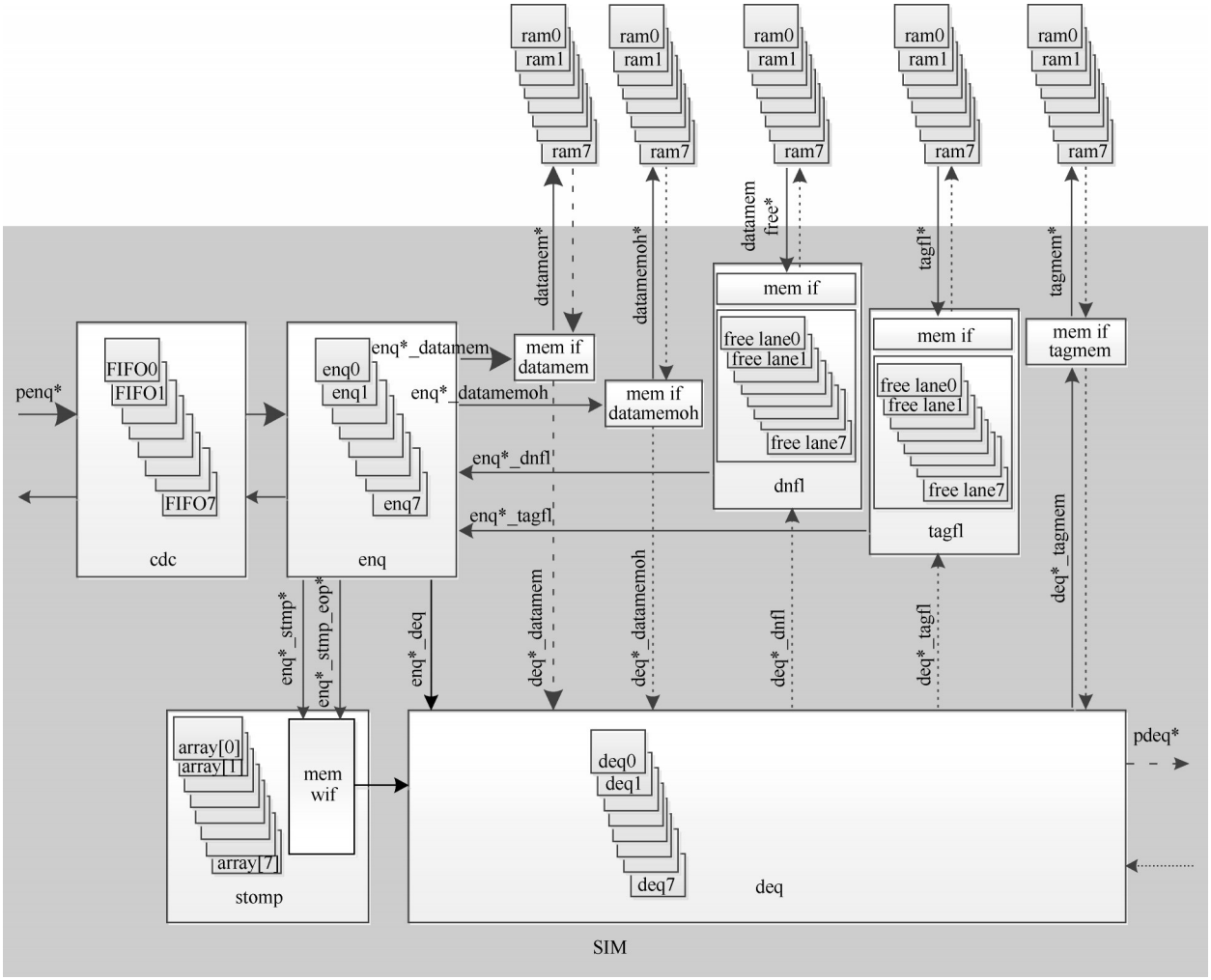


图3　SIM结构

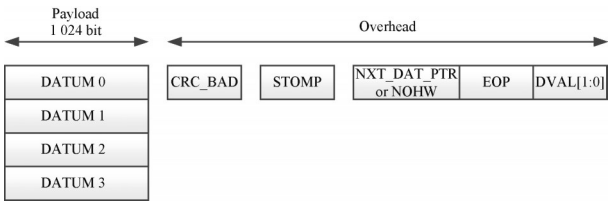


图4　Datanode和Overhead信息的数据结构

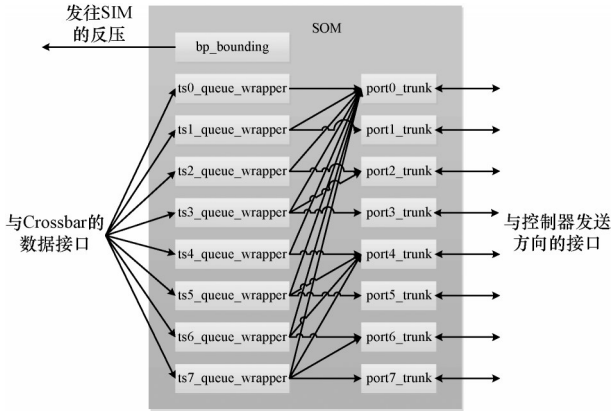


图5　SOM结构

2.2　共享缓存输出结构设计

SOM作为流量分离器将来自Crossbar互连网络的数据分发至n个输出端口。SOM结构如图5所示，其中n=8。

在SIM中，为避免头阻塞，基于源端口与目的端口建立多组VOQ，为避免死锁及支持服务质量（QoS, quality of service），每组VOQ又细分为多条优先级队列。同样地，在SOM中，也基于优先级建立多条队列，用于实现QoS支持及相同优先级报文的保序输出。

SOM接收到来自第二级Crossbar交换的报文后，会对其进行基于优先级的缓存管理与队列管理。为利于调度以及便于绑定模式下端口间缓存融合的实现，缓存管理与队列管理相分离。无论是处于绑定模式还是非绑定模式下，端口与队列管理间

的对应关系不变，但端口与缓存管理间的对应关系则不固定，非绑定模式下端口对应的缓存仅为端口本身缓存，绑定模式下端口对应的缓存则为端口本身缓存与其他无效端口对应的缓存根据绑定模式进行逻辑融合后的缓存。简而言之，缓存管理与队列管理实现逻辑上的分离，便于端口间缓存融合处理逻辑的实现，而缓存融合处理逻辑的实现，利于缓存资源的充分利用。

本文设计的交换电路通过 SIM 和 SOM 与外部不同协议控制器和模块连接，外部协议控制器和模块的工作时钟域存在与交换模块的工作时钟域不相同的情况，交换需要根据情况进行跨时钟域处理。跨时钟域的实现方面，SIM 借助专门用于跨时钟域的异步先进先出（FIFO），而 SOM 则复用内部功能 FIFO 以降低跨时钟域处理带来的逻辑与缓存资源消耗。

2.3　基于时隙的仲裁调度机制设计

CSOQ 结构属于 Time-Space-Time 结构，两端的 Time 对应 PG 输入和输出方向的总线时分复用，Space 代表核心 Crossbar 交换结构输入和输出端口空间上的匹配。仲裁调度的目的是在尽量达到最大转发效率的前提下，确定输入端口到输出端口的连接关系。本文设计如果直接采用端口对端口的仲裁调度，则需要在达成输入和输出端口最大数匹配的同时考虑总线时分复用导致的冲突，调度的实现面临时间与空间交织的挑战。为此，本文引入时隙概念，将 CSOQ 结构的调度过程进行时间和空间的解耦，主要分为下述两步。

1) 在每个 PG 中，将目的端口相同的队列请求进行合并，使得最初以源端口为区分维度的调度请求转换为以 PG 为区分维度统一输出。

2) 将一个调度周期以单时钟周期为粒度分为 n 个时隙，在 T_i（i 取值范围为 0~n–1）时隙仅处理目的端口为各个 PG 中 P_i 端口的调度请求。实现上设置 n 个集中调度模块，分别在 n 个时隙锁定仲裁结果。如此，若每个报文的长度均为一拍（对应单个时钟周期），则 n 个时隙结束之后，每个输出端口都参与一次调度，确保了输出端口的调度公平性。分时隙调度策略如图 6 所示，其中 n=8。

通过上述两步，可将输出方向时间上的复杂度拆解到 n 个时隙内，同时将输入方向上的时间调度放在 PG 内部解决，而中间 Crossbar 的调度只考虑实现 PG 间的匹配。

PCIe、Ethernet 和 RapidIO 等协议具有绑定特性，当协议端口进行绑定后，首先带来的变化是交换端口带宽需求的成倍增加，增加的倍数取决于参与绑定的通道个数，带宽的增加导致需要更大的存储来进行数据的缓存。因此，PG 内端口的缓存与带宽资源分配需要能够适配绑定模式的变化，即非绑定模式下各个独立端口的缓存与带宽在绑定模式下需要能够根据绑定模式进行“融合”后重新分配给绑定的端口。绑定模式下缓存的再分配即前文所述 SIM、SOM 中的缓存根据绑定模式进行逻辑上的融合；绑定模式下带宽的再分配则基于时隙在逻辑上的融合来实现，即根据目的端口的绑定模式，对源端口的调度请求进行绑定处理，使得处于绑定模式的目的端口由原来非绑定模式下只在一个时隙参与调度变为在多个时隙参与调度，如此实现其带宽与绑定模式的对应。

3　仿真与分析

3.1　仿真环境与参数设置

在自研的一款异构协议软件定义互连芯片开发过程中，对本文所述交换电路进行了编码实现，并基于通用验证方法学（UVM, universal verification methodology）和 VCS（Synopsys 公司编译型 Verilog 模拟器）的仿真环境对设计的电路进行了仿真评估。参数设置情况为：最大交换规模 64×64，数

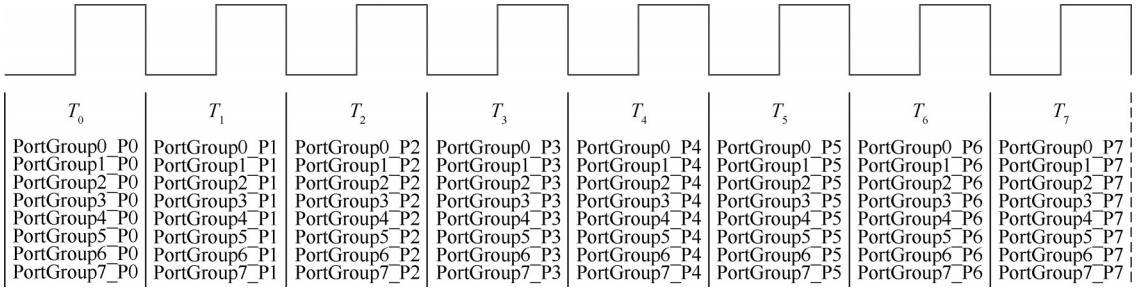


图 6　分时隙调度策略

据总线位宽1 024 bit，工作时钟频率800 MHz，核心Crossbar交换规模8×8，每8个端口组成一个PG，共享一条带宽800 Gbit/s的内部总线。PG内共享输入缓存3 072 KB，缓存管理粒度512 B，端口组内共享输出缓存560 KB。为评估本文设计的有效性，对不同绑定模式下交换端口的吞吐量、缓存、缓存利用率进行了仿真分析，通道绑定模式遍历了1×、2×、4×、8×，报文长度选取了8~4 096 B范围内的若干典型值。此外，对交换电路的资源消耗、时延性能以及预期性能的获取所带来的代价也进行了分析。

3.2　仿真结果分析

端口缓存方面，CSOQ结构能够弹性适应绑定模式的变化。不同绑定模式下CSOQ与CIOQ结构接收缓存存储报文个数如图7所示。从图7可以看出，CSOQ结构接收缓存存储的报文个数随绑定模式成比例变化，参与绑定的通道数越多，绑定端口所能存储的报文个数越多，表明CSOQ结构能够按需适应绑定模式变化引起的交换端口缓存需求变化。传统CIOQ结构下的端口缓存并不随绑定模式的变化而变化，而是始终提供最大绑定模式下的缓存能力，当端口处于非最大绑定模式时不可避免地存在缓存资源浪费。

相似地，端口带宽方面，CSOQ结构也能够弹性适应绑定模式的变化。不同绑定模式下CSOQ与CIOQ结构最大吞吐量对比如图8所示。从图8可以看出，CSOQ结构最大吞吐量基本随绑定模式成比例变化，参与绑定的通道数越多，绑定端口所能达到的吞吐量就越大，表明所设计的多协议交换电路能够按需适应绑定模式变化引起的交换端口带宽需求变化。同时可知，交换端口最大吞吐量能达到约800 Gbit/s，能够满足异构协议1.25~256 Gbit/s的端口速率需求。传统CIOQ结构下的端口带宽并不随绑定模式的变化而变化，而是始终提供最大绑定模式下的带宽能力，当端口处于非最大绑定模式时不可避免地存在带宽资源浪费。

CSOQ结构中端口缓存与端口带宽能够根据绑定模式适应性变化的弹性特性的实现，依靠的底层逻辑是PG内端口间缓存和带宽能够根据绑定模式按需拆分与融合，这有助于提高端口缓存与端口带宽的利用率。

基于本文设计，CSOQ与传统CIOQ结构数据最高缓存利用率如图9所示。由图9可知，在1×、2×、4×、8×情况下，当报文长度为缓存管理粒度的整数倍时，CSOQ结构端口的数据最高缓存利用率均能达到100%，如此高的缓存利用率的实现依靠的是第一级共享缓存交换内端口间缓存的按需分配与共享机制，以及基于Datanode的缓存管理机制。而传统CIOQ结构不同绑定模式下端口的最高缓存利用率仅为12.5%，原因分析如下。对于支持1×、2×、4×、8×绑定模式的传统CIOQ，由于端口不具有弹性，当物理端口存在N个8×端口时，交换电路就要支持8N个端口，且对称性要求每个端口需按照8×模式进行缓存设置。当端口全部为8×

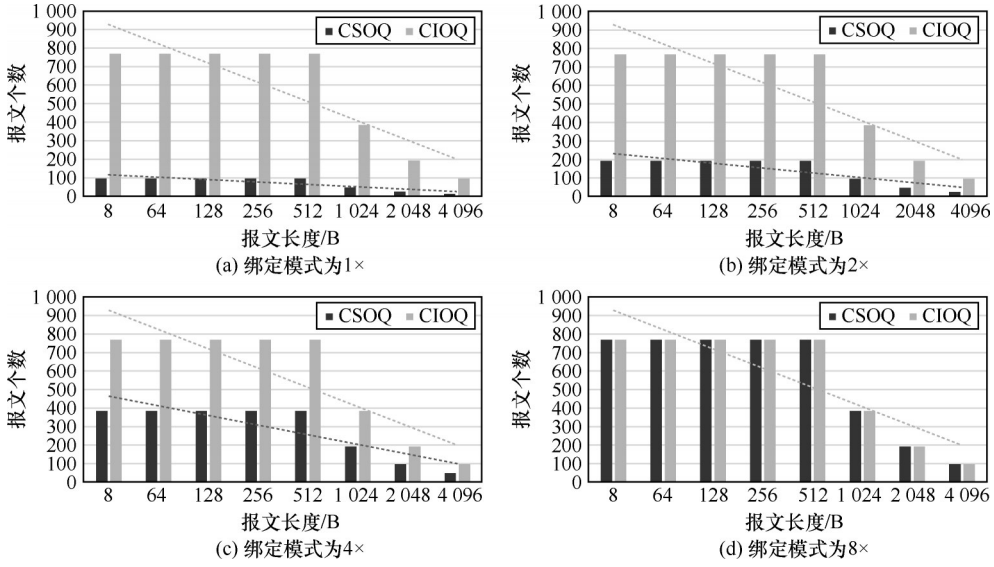


图7　不同绑定模式下CSOQ与CIOQ结构接收缓存存储报文个数

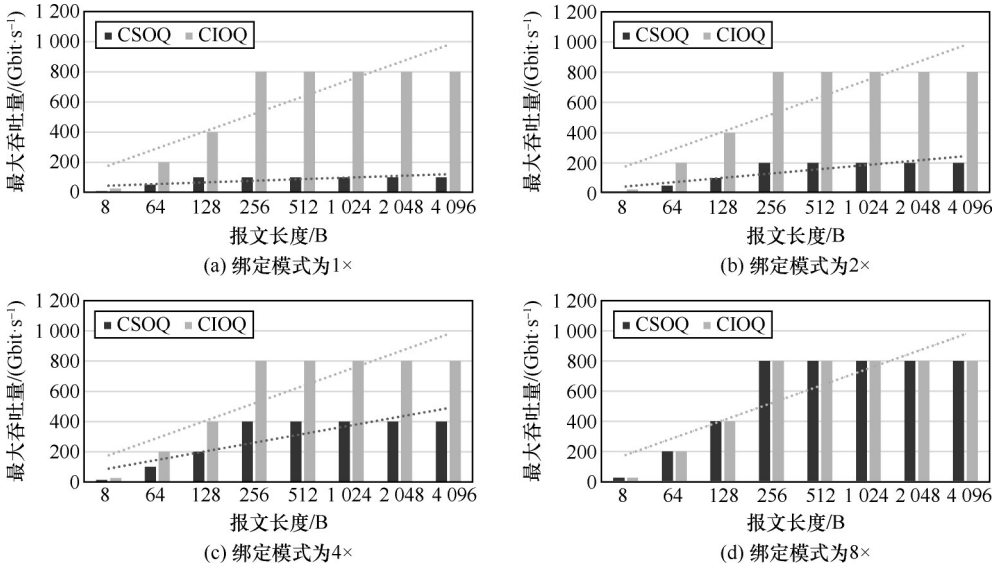


图8　不同绑定模式下CSOQ与传统CIOQ结构最大吞吐量对比

模式时，交换电路只有N个端口工作，不工作的端口所对应的缓存将暂时闲置，造成浪费；当端口全部为1×模式时，即使8N个端口都在工作，但交换端口实际所需缓存只是8×模式的$\frac{1}{8}$，同样会造成浪费。

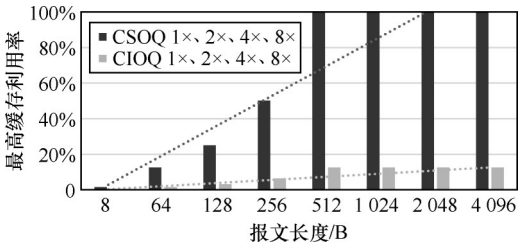


图9　CSOQ与传统CIOQ结构数据最高缓存利用率

可以看出，本文设计因支持不同绑定模式下端口间缓存的按需拆分与融合，在缓存利用率方面优势明显。在1×、2×、4×、8×模式下，缓存利用率最高均能提高87.5%，这意味着相同交换规模下，缓存资源开销约为传统CIOQ交换的$\frac{1}{8}$，这会带来面积与功耗方面的优势。表2为在相同交换规模（8×8）、相同端口性能情况下，基于本文设计、传统CIOQ、CSCQ（combined shared memory and crosspoint queuing）[3]架构的交换电路的FPGA资源开销及占比。从表2可以看出，本文设计的逻辑资源开销稍多，原因是本文设计支持的针对通道绑定的时隙融合与缓存融合功能提升了设计复杂度，但在缓存资源开销方面，由于缓存利用率较高，本文设计缓存资源开销明显较小。

表2　FPGA资源开销及占比

结构	LUT/个	FF/个	LUTRAM/个	BRAM/个
本文设计	999 013 (24.45%)	549 048 (6.72%)	10 504 (1.10%)	777.50 (36.00%)
传统CIOQ	978 992 (23.96%)	521 269 (6.38%)	75 438 (7.90%)	885.49 (41.00%)
CSCQ	997 379 (24.41%)	512 281 (6.27%)	76 966 (8.06%)	928.68 (43.00%)

时延方面，本文设计能够提供数十纳秒的转发时延，与传统CIOQ交换时延性能相当，CSCQ交换时延性能略好。表3为在本文设计、传统CIOQ、CSCQ交换架构下，长度为64 B的报文在无阻塞情况下的交换转发时延。结果显示，输入跨时钟域逻辑非旁路情况下，本文设计的总时延为28~(28+n–1)个时钟周期，其中，n–1由第一级共享缓存的时分复用特性引入。相比于CSCQ，低转发时延的实现受益于第二级Crossbar交换架构中交叉节点缓存的取消。图10为本文设计输入跨时钟域旁路情况下的最低交换转发时延仿真波形，在800 MHz时钟频率下，转发时延为21×1.25 ns=26.25 ns。本文设计能够满足RapidIO、FC、Ethernet、PCIe这4种异构协议的转发时延需求。

本文设计内部多处需要根据不同绑定模式进行多路选择，在数据总线较宽、绑定模式较多的情况下，会导致后端布线拥塞，后端设计难度增加，因而对后端设计要求较高。这是获得高灵活性、高资源利用率特性，以及具备最大限度的多协议交换应用灵活性的代价。

表3　交换转发时延

结构	输入跨时钟域/个时钟周期	入队与仲裁输出/个时钟周期	总时延/个时钟周期
本文设计	7（含2外部时钟）	21~(21+n−1)（含5外部时钟）	28~(28+n−1)（含7外部时钟）
传统CIOQ	7（含2外部时钟）	20（含5外部时钟）	27（含7外部时钟）
CSCQ	7（含2外部时钟）	24~(24+n−1)（含5外部时钟）	31~(31+n−1)（含7外部时钟）

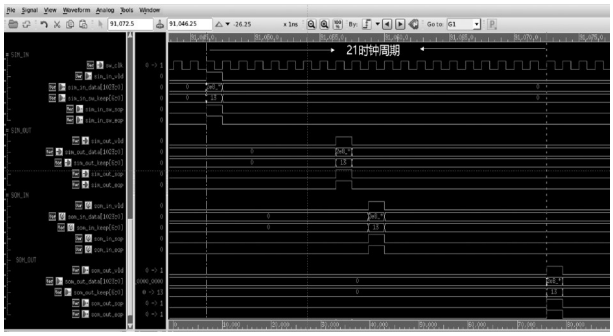


图10　最低交换转发时延仿真波形

表4为4种交换电路结构的主要设计指标对比，分别是一款商业交换芯片RXS2448、文献[3-4]和本文设计。RXS2448采用的是适配性修改后的Crossbar交换架构，端口缓存及带宽按照2×模式设置，1×模式下存在缓存及带宽浪费情况，仅2×绑定为4×时进行缓存与带宽的融合，可以看出，本文设计支持的绑定模式更多，支持的端口速率范围更广；相比于文献[3]，两者在架构方面比较相似，均为共享缓存+Crossbar交换架构，区别是本文设计采用的是中间节点不带缓存的Crossbar，在时延及缓存需求方面占有优势；相比于文献[4]，本文设计通过在共享缓存交换架构的基础上引入Crossbar交换架构对端口进行分组，克服了单一共享缓存交换架构无法提供超低转发时延的缺点。整体来看，相比于参考设计，本文设计能够弹性适应绑定模式的变化，在绑定模式、端口速率以及报文长度等方面支持的范围更广，在缓存利用率方面表现得更好，在应用范围方面不局限于某一单一协议交换，能够应用于RapidIO、PCIe、FC、Ethernet这4种单一协议交换及多种协议组合的混合协议交换。

4　结束语

不同的交换架构具有不同的交换特性，适用于不同的协议类型，多协议交换架构因面向多种异构协议，所以需要兼备不同的交换特性。本文针对软件定义互连系统中异构协议交换需求，提出了一种共享缓存与Crossbar相融合的两级交换架构CSOQ，CSOQ通过2种交换架构的融合实现对多种交换特性的兼顾。另外，针对CSOQ架构时间空间耦合的特性，本文提出了一种基于时隙的多级调度方法，实现了对时间空间的解耦，简化了调度实现。最后，通过仿真分析，本文对所提交换电路的性能优点进行了验证，证明该架构及对应调度方法在异构协议支持方面具有显著优势。后续还将在协议扩展、调度算法等方面进行探索与优化，以适应软件定义系统更广泛和更简便的应用。

表4　4种交换电路结构的主要设计指标对比

主要设计指标	交换架构	支持的绑定模式	支持的端口速率/(Gbit·s^{-1})	支持的最大报文长度/B	缓存需求	缓存利用率	能否根据绑定模式弹性调整端口数量、带宽、缓存	最大交换规模	面向的协议类型
RXS2448	Crossbar（中间节点带缓存）	1×, 2×, 4×	1.25~50.00	280	中间节点缓存容量随交换规模指数增长	1×模式，最高达50%；其他模式，最高达100%	能（比较有局限性）	24×24	一种：RapidIO
文献[3]	共享缓存+Crossbar（中间节点带缓存）	1×	10.00	未明述	中间节点缓存容量随核心Crossbar规模指数增长	1×模式，最高达100%	不能	32×32	一种：Ethernet
文献[4]	共享缓存	1×	0.1~10.00	1 518	随交换规模线性增长	1×模式，最高达100%	能	9×9	一种：Ethernet
本文设计	共享缓存+Crossbar（中间节点不带缓存）	1×, 2×, 4×, 8×等	1.25~256.00	4 096	随交换规模线性增长	不同绑定模式下，最高均可达100%	能	64×64	4种：RapidIO、PCIe、FC、Ethernet

参考文献:

[1] 吕平, 刘勤让, 邬江兴, 等. 新一代软件定义体系结构[J]. 中国科学: 信息科学, 2018, 48(3): 315-328.
LYU P, LIU Q R, WU J X, et al. New generation software-defined architecture[J]. Scientia Sinica (Informationis), 2018, 48(3): 315-328.

[2] AL-BAWANI K, ENGLERT M, WESTERMANN M. Online packet scheduling for CIOQ and buffered crossbar switches[J]. Algorithmica, 2018, 80(12): 3861-3888.

[3] ZHENG L, PAN W T, GAO Y, et al. Architecture design and performance analysis of a novel memory system for high-bandwidth onboard switching fabric[J]. Computer Networks, 2021, 198: 108367.

[4] 付文文, 刘汝霖, 全巍, 等. nPSA:一种面向TSN芯片的低延时确定性交换架构[J]. 计算机研究与发展, 2023, 60(6): 1322-1336.
FU W W, LIU R L, QUAN W, et al. nPSA: a low-latency, deterministic switching architecture for TSN chips[J]. Journal of Computer Research and Development, 2023, 60(6): 1322-1336.

[5] WU C C, QIAO L F, CHEN Q H. Design of a 640-gbps two-stage switch fabric for satellite on-board switches[J]. IEEE Access, 2020, 8: 68725-68735.

[6] DEVI R C, FLORINABEL D J, PRASANTH N. High throughput scheduling algorithms for input queued packet switches[J]. Computers, Materials & Continua, 2022, 70(1): 1527-1540.

[7] LI Z H, WAN H, DENG Y D, et al. Time-triggered switch-memory-switch architecture for time-sensitive networking switches[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2020, 39(1): 185-198.

[8] 尘福兴, 李挥, 崔凯, 等. 基于布尔集线器的线速多播自路由交换结构[J]. 通信学报, 2014, 35(7): 129-139.
CHEN F X, LI H, CUI K, et al. Novel wire-speed multicast self-routing switch fabric based on Boolean concentrator[J]. Journal on Communications, 2014, 35(7): 129-139.

[9] 黄慧群, 刘勤让, 卜佑军, 等. 交换结构中的可重构缓存机制[J]. 通信学报, 2012, 33(10): 126-131.
HUANG H Q, LIU Q R, BU Y J, et al. Reconfigurable buffer mechanism in switch fabric design[J]. Journal on Communications, 2012, 33(10): 126-131.

[10] ZHENG L, YI C, PAN W T, et al. Combined shared-memory and buffered-crossbar architecture for high-bandwidth onboard switching fabric[C]//Proceedings of the 5th Asia-Pacific Workshop on Networking. New York: ACM Press, 2021: 3-4.

[11] PAN B Y, ZHOU Q Q, CHEN G. CQPPS: a scalable multi-path switch fabric without back pressure[J]. IET Communications, 2021, 15(16): 2036-2045.

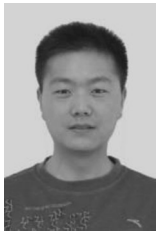
[12] HUANG S J, WANG M W, CUI Y. Traffic-aware buffer management in shared memory switches[J]. IEEE/ACM Transactions on Networking, 2022, 30(6): 2559-2537.

[13] ZHENG L, QIU Z L, PAN W T, et al. Analysis of a shared-private buffer management scheme for shared memory switches[C]//Proceedings of the 2018 International Conference on Computer, Information and Telecommunication Systems (CITS). Piscataway: IEEE Press, 2018: 1-5.

[14] 张卜方, 刘淑涛, 刘丙亚, 等. 一种基于CICQ结构的比例公平调度算法[J]. 中国集成电路, 2022, 31(7): 48-52.
ZHANG B F, LIU S T, LIU B Y, et al. A proportional fairness scheduling algorithm based on CICQ switch[J]. China Integrated Circuit, 2022, 31(7): 48-52.

[15] QIAO S Y, HU C C, BREBNER G, et al. Adaptable switch: a heterogeneous switch architecture for network-centric computing[J]. IEEE Communications Magazine, 2020, 58(12): 64-69.

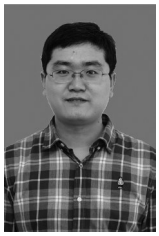
[作者简介]



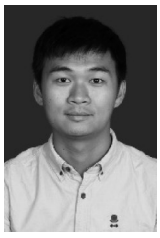
董春雷（1987– ），男，河南周口人，信息工程大学助理研究员，主要研究方向为高速交换芯片设计、软件定义互连技术。

沈剑良（1982– ），男，浙江德清人，博士，信息工程大学副教授，主要研究方向为新型网络体系结构、高速接口设计。

李沛杰（1990– ），男，山西襄汾人，博士，信息工程大学助理研究员，主要研究方向为高速接口设计、软件定义互连技术、现代SoC设计技术。



王盼（1984–），男，河北石家庄人，天津市滨海新区信息技术创新中心高级工程师，主要研究方向为交换芯片系统架构。



薄光明（1987– ），男，河南周口人，天津职业技术师范大学工程师，主要研究方向为网络信息安全技术、实验室建设与安全管理。

路凯（1999– ），男，山东临沂人，信息工程大学硕士生，主要研究方向为网络交换芯片设计。